University of Sheffield

Hybrid Model of Ballistics

Haolin Wang

Supervisor: Dr Ramsay Taylor

A report submitted in fulfilment of the requirements
for the degree of MSc in Data Analytics

in the

Department of Computer Science

September 14, 2022

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Haolin Wang

Signature: Haolin Wang

Date: 13/09/2022

Abstract

Firearms are used for a variety of purposes, including self-defense, hunting, security guarding, competitions, and ancient and modern warfare. Ballistics, the study of the motion of projectiles, has been explored by humans since the invention of firearms, which may be traced to the mediaeval times. Existing mathematical expressions can determine motion, but in practise, the shooting environment may be complicated, and a law of motion model based on ideal conditions may not be accurate. A simple question is raised: Is the existing estimating approach accurate, and if not, how could it be modified to better match to real-world data? This project is to answer the problem of estimating velocity of bullet at varied distances from the muzzle, depending on the characteristics of bullet. The dataset for this project will be analysed using the equation of motion, together with symbolic regression, a machine learning technique that creates a mathematical expression. Genetic programming and the DEAP framework have been selected as analysis tools for symbolic regression. The aim of this project is to train a model that fits the datasets from Ammo & Ballistics 6[14].

A hybrid model based on a combination of the equation of motion and symbolic regression is developed and demonstrates a relatively good performance not only on the ammunition type for training, but also on test sets containing data from bullet models somewhat distinct from the training and validating sets, which may indicate that the model has a good level of generalisation.

A model on internal ballistics is trained, which predicts the initial velocity of a bullet at the muzzle depending on bullet parameters and gunpowder weight. The relation of training dataset size and genetic programming model performance is also examined.

Acknowledgements

First, I would like to express my deepest gratitude to my supervisor, Dr. Ramsay Taylor, for proposing such an exciting project, lending me the ballistics books, and being so helpful and encouraging throughout my dissertation. I also want to thank University of Sheffield, for offering me the place to complete my study, fulfilling my academic needs and after this year I could proudly say that I am well prepared for my next chapter after graduation.

I'd want to thank everyone in Allen Court B10, as well as the many other friends I've met in Sheffield, for making each and every day something to look forward to.

Finally, I would like to thank my family for always being supportive of my every single decision and for encouraging me overcome each difficulty.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There is a wide range of application of firearms in self-defencing, hunting, security guarding, ancient and modern warfare and competitions. The ballistics, a science that studies the motion of projectiles has already been explored by people since the advent of the firearm, which may be traced back to the middle ages. A precise and quick prediction of the trajectory is the key to enhancing shooting accuracy, as well as the premise and guidance for adjusting the firearm. Using numerical integration to solve the ballistic equation is the conventional method for trajectory prediction. However, achieving a precise solution value requires a minor integration step and a complicated trajectory model, which will definitely increase both processing time and computational complexity. The difficulty of rapidly and effectively determining the bullet trajectory has emerged as a subject worthy of consideration.

Advanced technologies have resulted in the development of tools for simulating ballistics, including ballistic trajectory charts, ballistics calculators, relevant measure equipment *etc.*. All of these techniques, nevertheless, have some drawbacks: First of all, since all of these techniques are restricted to already-existing bullet types, it is almost impossible to predict any future design without production and experiments, which not only increases the time required but also the expense of the research. Moreover, none of the aforementioned methods provides a specific function that describes the movement of bullets, which would be essential if we wished to further investigate the trajectory or extrapolate the initial condition of the projectile based on a particular state of the bullet, such as determining the initial condition of a bullet provided its velocity at a certain distance from the muzzle. A mathematical formulation of bullet trajectory can also be used to explore the effect of a particular property(*e.g.* bullet tip shape) on the velocity of bullet. Hence, the purpose of this research is to demonstrate a machine learning technique for problem prediction by returning a function that reflects bullet motion based on current ballistic data. In the future, if the function has a high degree of precision, it will be of great reference for trajectory calculations and engineering design.

In this dissertation, the ballistic coefficient and bullet weight were used to model the motion of certain bullet models. Three types of trajectory prediction were established, one is derived from the equation of motion, one is from a symbolic regression method based on genetic programming. A hybrid model that incorporated the two methods mentioned above was also

considered. Regarding the internal component of ballistics, I explore the correlation between the specifications of bullet(*e.g.* diameter, weight, tail shape, tip shape) with the amount of gunpowder loaded and the initial speed of the bullet at muzzle. These methods mentioned above will be discussed further in Chapter 4.

1.1 Overview of the Report

Chapter 2 gives a short literature study for this project, including an introduction to ballistics, the mathematics involved in ballistics, and elements that may impact the velocity of a projectile throughout its travel. Regression techniques in machine learning are discussed briefly, along with the differences between symbolic regression and other classic regression approaches. In this chapter, the justifications for employing symbolic regression in this project and for selecting genetic regression to solve symbolic regression problems are outlined. The ballistic calculator, an existing tool for modelling ballistic trajectories, is briefly presented and its limitations are highlighted.
Chapter 3 covers the selection of datasets, explains the bullet types employed for the train, validation, and test sets, and provides a brief introduction to the methodologies that will be used in subsequent chapters. The ethical issues are also addressed in this chapter.
Chapter 4 provides a comprehensive project plan. Each method contained in this project is described in detail. In this chapter, the tools and algorithms used are thoroughly discussed.
Chapter 5 discussed experiment outcomes. RMSE is used to evaluate the performance of each approach by measuring the error value from the dataset of each method. The process of hyperparameter tuning and the performance of the model at various distances are also covered in this chapter.
The conclusion of this project is stated in chapter 6.
The results are detailed in the appendices. The model of symbolic regression is presented in appendix A, whereas the specifications of the hybrid model are presented in appendix B.

Chapter 2

Literature Survey

This chapter introduced the project's theoretical foundation. Including ballistics and important external ballistics measures, as well as a tool for computing external ballistics. This chapter also introduces symbolic regression, the machine learning technique that will be utilised in this project, and explains its benefits.

2.1 Ballistic

2.1.1 Overview of Ballistics

Ballistics is typically divided into four categories[44]:

- **Interior ballistics** studies the projectile propulsion. In guns, internal ballistics refers to the movement from the ignition of the propellant and the projectile's escape from the gun barrel.
- **External ballistics** studies the period from the time the projectile leaves the muzzle and moves through the air until it hits the target.
- **Transitional ballistics**(also called intermediate ballistics) studies the behavior of the projectile from the moment it leaves the muzzle until the pressure from propellant is subsided.
- **Terminal ballistics** studies the movement and behaviour of bullet after it hits the target.

In this project, I focus on mainly exterior ballistic. External ballistics and the factors that affect external ballistic forces will be discussed in details in next session. An experiment on internal ballistics is done and discussed in chapter 4 and 5, but the mechanism of internal ballistics is not discussed, as the factors that may affect the propelling force when bullet is in the barrel can be various, the type, quality and volume of propellant, the shape of cartridge chamber and the mass and shape of bullet, and the chemistry and complex mechanics of the burning process will not be discussed in this dissertation, I'll only investigate on the

connection between loaded gunpowder volume and initial velocity of bullet at the muzzle for internal ballistics.

2.1.2 External ballistics

External ballistics is the study of the movement of projectiles after they have left the barrel. The varieties of projectiles may range from those of 4mm calibre and half a gramme to those of 9 metres in length and 16 tonnes. The size and weight of a projectile significantly affect its trajectory; more specifically, the amount of air resistance it encounters is highly influenced by its dimensions. The earliest known firearm was invented in China in the 10th century, where it was able to shoot out flame horizontally out of a hollow bamboo tube loaded with rocket[41]. However, not until the late 15^{th} century guns reached their "classic" form, where guns became longer, lighter, more productive and more accurate, and the study about the bullet movement became more reasonable. Isaac Newton is one of the modern founders of external ballistics, as Newton's laws of motion established the framework of modern classical mechanics, without which ballistic would not become a science. The trajectory of bullet is determined by two types of forces: gravity and aerodynamic forces. Gravity depends on the mass of the projectile and acts downwards in the vertical direction, causing the projectile to fall below the line of sight after a long distance travel. Major aerodynamic forces include normal force, drag force, lift force, pitching moment, Magnus moment, and pitch damping moments(Lahti *et al.*, 2019)[27]. This dissertation focuses on the bullets used in firearms, or guns, which is designed to be easily carried and used by individuals. The majority of aerodynamic forces for a bullet fired from a gun with a flat trajectory can be neglected due to the small travel distance and light weight of the projectile; these forces are minor compared to the drag force. In this project the equation of motion is used as the foundation of theoretical approach of solving the trajectory problem. An equation of projectile speed can be deduced from the Newton's second law of motion($F = m \cdot a$)[33]:

$$m\frac{d\overrightarrow{V}}{dt} = \sum \overrightarrow{F} + m\overrightarrow{g} + m\overrightarrow{\Lambda} \tag{2.1}$$

$\sum \overrightarrow{F}$ in (2.1) refers to the summation of all external forces the projectile experiencing through the movement, which is a summation of drag force and force from crosswind in this project.

2.1.3 Ballistic Coefficient

Ballistic coefficient illustrates the extent of a projectile in flight is slowed down by air resistance, it is affected by the projectile's mass, drag coefficient, and cross sectional area in the direction of its motion in relation to the atmosphere[29]. It plays a significant role in both the analysis of the performance of the bullet and the engineering design. In comparison to a bullet with a low ballistic coefficient, a high ballistic coefficient bullet will travel farther. A high ballistic coefficient bullet will shoot flatter, maintain its initial speed better, and have

better resistance at the wind.
Ballistic coefficient can be calculated by measuring the velocity of bullets at different distance from muzzle[9][42]. When selling ammo, the manufacturer will provide a reference value of ballistic coefficient. According to the research from Courtney and Courtney(2007)[12], many bullet manufacturers are likely to overstate ballistic coefficient values for marketing purpose, and their experiment shows there is an exaggeration from 5-25% among the majority of fourteen bullets with calibre in range 0.224-0.308 inches. If we assume that the ballistic coefficient value is too optimistic or exaggerated, a series of questions come to mind: Is the theoretical approach of trajectory(e.g. equation of motion) using published ballistic coefficient value still reliable? Is there a more accurate technique to simulate the trajectory of a bullet that depends more on other feature of it, such as weight, material and design? This will be considered and further discussed in chapter 5, which discussed both models based on ballistic coefficient and on mass and diameter only.

2.1.4 Drag coefficient

Drag coefficient is a quantity that describes the resistance of an object from a fluid environment. A low drag coefficient represents that the object receives less aerodynamic drag force from the environment, such as "streamlined" design, which is known for its low air resistance. Drag coefficient associates with the speed of object, density of surrounding fluid, and the shape of object.
In ballistics, the term "drag coefficient" usually refers to the standardised drag models. There are several types, but the G1 and G7 are the two most frequently used by people. G1 and G7 are two standard aerodynamic drag models with different projectile shape, where G1 is more preferable for flat-based projectile and G7 is designed for a longer, boat tailed bullet. Ballistic coefficient of a projectile is calculated as $C_{\mathrm{BC}} = \dfrac{m}{C_{\mathrm{d}} A}$, where m is the mass of projectile,C_d the drag coefficient of the standard model, A the cross sectional area of the projectile and C_{BC} refers to the ballistic coefficient related to a standard drag model(*e.g.* G1 ballistic coefficient).
There are a few scientific studies published on the relation of projectile shape and drag coefficient: Rafeie and Teymourtash(2016)[38] studied three shapes(wadcutter, sharp-pointed, round nose) of 4.5mm calibre by finding a numerical solution of the Navier-Stokes equations, their analysis showed the round nose pellet has the best aerodynamic performance while the sharp-pointed pellet has flattest trajectory(*i.e.* fastest initial velocity). Their analysis also indicated a sharp rise follows by a slow decrease of drag coefficient when speed goes from subsonic to supersonic for all three bullet shapes. Salimipour *et al.*[40] carried out a more comprehensive and accurate simulation on 4.5, 5.5 and 6.35mm projectiles in 2018 with Naiver-Stokes equations, approved Rafeie and Teymourtash's findings, and showed a flat-nose bullet has least attitude loss in short distance(10m), while round-nose bullet performs better in flatness at a longer distance(35m). Ladommatos[25][26] tested the relationship of different bullet shape and by detailed experiments and obtained a similar result of round-nose less than sharp-nose less than flat-nose in drag coefficient, but were limited to only at speeds

below 0.57 Mach. These studies inform my decision to incorporate bullet shape and design as variables during machine learning model training.
The drag coefficient often changes significantly when an object exceeds the speed of sound, and as the movement of bullet usually start with a high initial velocity, this change of drag coefficient matters in ballistic study. The outcome from Chartes and Thomas indicates that when the velocity of a sphere from a supersonic to a transonic range, the drag coefficient will decrease. Through the transonic zone, the drag coefficient rapidly decreases, and at the subsonic zone, it nearly remains unchanged[10]. A large number of experiments on different calibres from Braun[7], McCoy[30] and Hitchcock[17] supported that bullets have similar behaviour on drag coefficient when bullet speed from supersonic to transonic.

2.2 Existing tool of modelling ballistic trajectories

Ballistic calculator is often used in predicting bullet trajectories. Contemporary ballistic calculators are usually based on mathematical models. An exterior ballistic calculator usually requires to select a bullet model from existence, ballistic coefficient, weight of bullet, initial velocity, shooting angle, and a series of features describe the environment of shooting, such as wind, temperature, altitude and humidity.

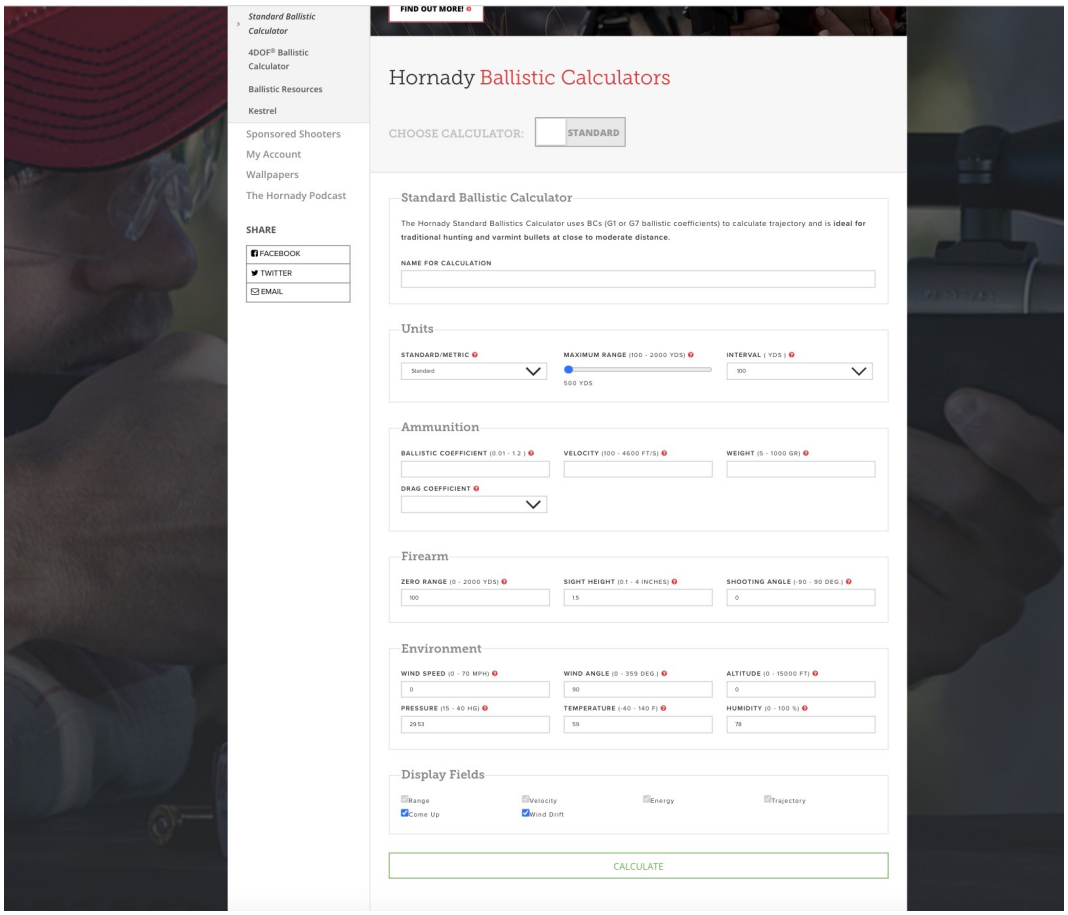


Figure 2.1: Example of a ballistic calculator, from Hornady[18]

Ballistic calculators use a standard drag force model and calculate the velocity and path bases on the mathematical functions that describes the whole system. The common models are G1 and G7, which illustrate different shape of bullet. This estimating method has its limitation: The drag force curves of bullets are more or less different from the G drag force curves since bullets rarely have the same shape as typical G models, which makes estimations never totally accurate.
In addition, existing ballistic calculators only accept certain bullet types from their database and need a variety of weapon and environment-specific parameters. If some parameters(*e.g.* wind speed, temperature) are inaccurate or unknown, it is almost difficult to determine an accurate trajectory using ballistic calculators.

2.3 Regression

Regression methods are used to evaluate the connection between a dependent variable and one or more independent variables. It is categorised as supervised machine learning. Regression

usually returns a specific mathematical rule with the given data. The most common forms of regression may be linear regression and logistic regression, with the former establishing a linear relationship between data and the latter modelling the likelihood that an event will occur. In this project, the symbolic regression technique is selected above all other traditional regression models, as the majority of which have the disadvantage of requiring a predefined data model structure. For example, in linear regression, people assume that the dataset has a linear correlation, and in logistic regression, the targets need to be binary or ordinal. In contrast to traditional regression approaches, symbolic regression avoids introducing presumptions and to instead presumes the model from the data, aiming to identify both the data correlation and model parameters.

2.3.1 Symbolic Regression

Symbolic regression is the technique of obtaining mathematical expressions that match actual output. It is an optimisation problem and often believed to be NP-hard[32]. Number of techniques are used to solve the challenges of symbolic regression, the popular methods are mainly based on genetic programming. Genetic programming is a iterative method that creates offspring from parent solutions by simulating the genetic crossover process and rejects undesirable solutions to evolve a group of candidate solutions. Koza[23] first presented the genetic programming model in 1992. In the "Koza-style" GP, the syntax tree made up of operators over inputs and constants optimised by the algorithm. The majority of symbolic regression research to date has originated from genetic programming sub-field.
There are other algorithms that are used to solve symbolic regression problem. Petersen *et al.*(2020)[37], Kim(2020) *et al.*[22] stated deep learning approaches of symbolic regression. The deep learning method has similar accuracy, but takes a longer time to approach to the right solution comparing to many genetic programming methods at most of the cases. Jin *et al.*[20] in 2019 is an example of applying the Markov chain Monte Carlo algorithm to aid in the search for a symbolic regression solution. Math expressions are represented by trees in the Jin *et al.*method, and new expression trees are generated based on past iterations, similar to genetic programming methods, while trees in each iteration operate "stay", "grow", "prune", "delete" and "insert" with certain probability, just as Monte Carlo algorithm.
La Cava *et al.*[24] purposed a standard framework for benchmarking symbolic regression methods and test the performance of 14 symbolic regression algorithms. Using a large real-world dataset from the Penn Machine Learning Benchmark (PMLB) and black-box regression tasks, La Cava *et al.*concluded that on real-world and black-box regression problems, GP-based symbolic regression approaches perform better than recent symbolic regression methods that depend on other fields, and on synthetic real-world physics and dynamical systems problems. This provides early suggestions for selecting symbolic regression through genetic programming as the project's methodology. The genetic programming algorithm is a standard and common technique for solving symbolic regression problems; hence, I have chosen to implement it in this project.

Chapter 3

Analysis

In this chapter, I explain in detail why 6.5mm bullets were chosen as the training data set, outline some of the constraints of this data set, and briefly describe the methodologies and formulas utilised in this project, which will be discussed in greater depth in later chapters.

3.1 Overview of dataset

The external ballistics data set was obtained from the book Ammo & Ballistics 6: For Hunters, Shooters, and Collectors by Forker[14], the data set is extensive given that it includes a wide range of bullets generally range from 4mm pistol cartridges to rifle cartridges for hunting. Here is one example from the book:

Hornady 120-grain GMX(81490)							G1 Ballistic Coefficient = 0.450		
Distance(Yards)	Muzzle	100	200	300	400	500	600	800	1000
Velocity(fps)	3050	2850	2658	2475	2298	2129	1968	1668	1409
Energy(ft-lbs)	2478	2163	1882	1631	1407	1208	1032	742	529
Taylor KO index	13.8	12.9	12.0	11.2	10.4	9.6	8.9	7.5	6.4
Path(Inches)	-1.5	1.4	0.0	-6.3	-18.3	-36.9	-63.2	-144.4	-276.0
Wind Drift(Inches)	0.0	0.6	2.5	5.8	10.6	17.1	25.6	49.3	83.6

Table 3.1: An example of 6.5mm Creedmoor from data set

This data set, but at the other hand, has a number of omissions:

- **Small number of data for each bullet type:**
  Although for each cartridge several sets of data is given, each set of data describes the motion of different bullet and produced by different brand. The number of data is small, which is usually not sufficient enough to train a machine learning model and may lead to a result of under-fitting. On the other hand, since the tests are being

carried out by humans, the data collected may fluctuate from the optimal outcome due to imperfections in the operating procedures and recording equipment. For example, a minor movement of the barrel during bullet release will add speed towards the projectile's initial muzzle velocity. Every piece of data contains a degree of error due to these unpredictable factors, and one of good approaches to remove the noise is to use a large quantity of information to eliminate the inaccuracy. If we could assume that the noise values are independent and random to each other, according to the Law of Large Number, the average of the observations acquired from a large number of trials should be near to the expected value and tends to become closer to the expected value as more trials are undertaken[13]. As a result, using current data set may lead to an outcome of inaccuracy, although the outcome may still acceptable, as in reality, shooting may subject to similar deviations.

- **Weight of bullet:**
  The weight of each bullet is stated in the data set measured in grains(grs), where 7000 grs is equal to 1 commercial pound in English units[6]. The data belonging to the same set (for example, 6.5 Creedmoor) have a similar shape as they fit the same cartridge, but the bullet designs may differ slightly in terms of size, material and weight. The weight of bullet is usually stated in the product name, such as Hornady 120-grain GMX(81490). Some bullets may be composed of different types of materials, such as the Hornady A-MAX, which has a polymer bullet tip and a metal body. Uneven weight distribution and varying centre of mass may impact the trajectory, but the dataset is insufficient to accommodate for this.

- **Propellant of cartridge:**
  A modern cartridge is usually consisting of four components: the case, the projectile, the propellant, and the primer. Each step of manufacturing process could affect the performance of the cartridge in use. The ammunitions under same cartridge are produced by different brand, which could clearly affect the performance of bullets. Performance variations are inevitable even with the same batch of bullets due to the manufacturing, shipping and experiment operating processes. For example, the condition of the propellant's combustion in the cartridge case may have an impact on the burn rate and, therefore, the bullet's propulsion energy. If we are simply thinking about the external ballistics, which begin to measure after the bullet exits the muzzle, then this may not be a deep concern.

- **Data of the environment:**
  Performance of bullet might be impacted by the shooting environment, such as temperature and humidity. As discussed in Chapter 2, the drag coefficient has a dramatic change when the speed of the bullet drops from about 1.2 Mach to below 1 Mach. The temperature has an obvious effect on the speed of sound in the air, with a formula[34]:

$$v_{sound} = 331.22 * (\frac{T + 273.15}{273.15})^{0.5} \tag{3.1}$$

Note that the equation 3.1 was originally provided in the unit of Kelvin(K) for the temperature and knots(kts) for speed. The equation 3.1 has been converted into units of Celsius($°C$) for temperature and metre per second for the speed for better understanding. From equation 3.1 we can see that at $0°C$ the sound speed is around $331m/s$, and when temperature increases to $30°C$ the sound speed increases to around $349m/s$, which means a 5.4% of increase. However, factors that raises the temperature of air around the bullet can be various: the propellant in cartridge not only brings the bullet power, but also heat through burning, heat generated by the friction between the bullet and the barrel, and the majority of the energy that is lost through resistance during projection will be converted to heat. The humidity of air has minor affect on the speed of sound, which is about 1.5 m/s difference between 0% and 100% humidity at standard pressure and temperature, yet, when temperature rises, the impact of humidity on sound speed will become more evident[46].
Moreover, burning propellant involves a chemical reaction. The starting temperature of the process influences the reaction's initial work rate, as it does with most chemical reactions.

3.2 Dataset selection

The 6.5mm diameter bullets data sets are chosen to use as train data in this dissertation. This includes the .260 Remington, 6.5mm Creedmoor and 6.5 Remington Magnum. 6.5 mm bullets are known for their relatively high sectional densities and ballistic coefficients, they were designed for long-range shooting and often have a good accuracy. Here is a comparison of three types of bullets that are frequently used, which are 6.5mm Creedmoor, .30-30 Winchester and .308 Winchester. All these three are rifle cartridges, have a similar shape design but in different size:

Cartridge	Model	Initial velocity (fps)	Weight (grs)	G1 Ballistic Coefficient
6.5mm Creedmoor	Hornady 120-grain GMX(81490)	3050	120	0.450
.30-30 Winchester	Remington 125-grain Core-Lokt Pointed Soft Point - Managed Recoil (RL30301)	2175	125	0.215
.308 Winchester	Winchester 120-grain Defender (S308PDB)	2850	120	0.256

Table 3.2: Comparison of three type of cartridges

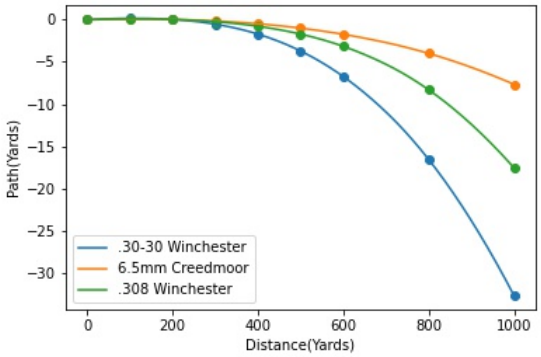


Figure 3.1: The path of three bullet travel by distance from muzzle

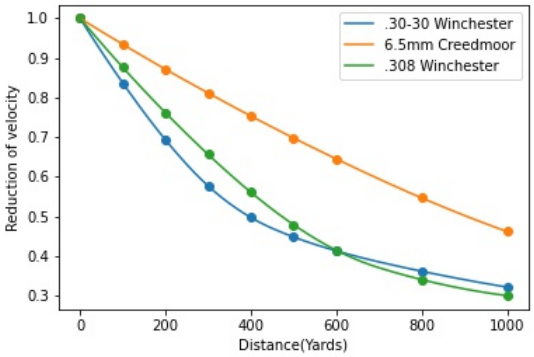


Figure 3.2: The reduction of speed of three bullet by distance from muzzle

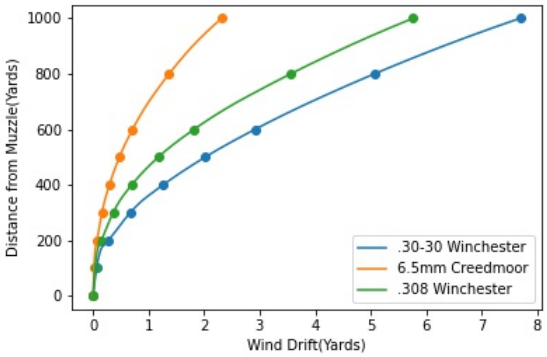


Figure 3.3: The wind drift of three bullet travel by distance from muzzle

For each bullet, there were only 9 data points provided in the data set. In order to plot smooth curves of trajectory, I use `interpld` from `scipy.interpolate` and choose 'quadratic' kind of interpolation. Figure 3.1 shows the extent of the bullet's descent in relation to the distance from the muzzle, and it can be observed that the 6.5mm Creedmoor has the smallest decline level among these three bullets. This indicates that the 6.5mm Creedmoor has a wider field of fire than the other two due to its faster speed. Figure 3.2 represent the rate of speed declined over the travel of bullet, these speeds were calculated by dividing them by the initial speed, as the initial speeds are various for different bullet. Additionally, as seen in Figure 3.3, the 6.5mm bullet has the least amount of wind drift out of the three types, lowering the uncertainty brought on by any external factors such as wind.
The data from book[14] includes the velocity, energy, Taylor KO index, vertical path and wind drift of the bullet at different distance from the muzzle. The velocity, vertical path, and wind drift are the three primary characteristics that explain the motion of the projectile in this study. Energy, which can be measured using velocity and mass, and the Taylor KO index, which determines the power of projectile at certain stage, are redundant in this project since they are less pertinent to the topic and can be easily calculated with the other three characteristics. The vertical path and wind drift of projectile was considered to be investigated together with velocity at first, but through coding I found that it is difficult to train the model with multiple outputs with current size of training set and equipment. Only the velocity of projectile is predicted in this project.

	Train	Validation	Test
Bullet type	6.5 Creedmoor & .260 Remington	.260 Remington	.260 Remington
			.280 Remington
			.338 Lapua Magnum
No. of Ammos	15	5	5

Table 3.3: Data chosen for train, validation and test

Table 3.3 shows the types of bullet used in train, validation and test sets. The machine learning model obtained from the 6.5mm calibre data set will be tested on different calibres. The bullet I choose are .260 Remington, .280 Remington and .338 Lapua Magnum. Some of the .260 Remington data is used in the training set, and the rest 5 bullets are used in a test set, in order to test the performance of model on a test set that is highly similar with the train set. .280 Remington uses slightly heavier bullet comparing to the training set, and have a greater calibre. .338 Lapua Magnum is quite different from the 6.5mm Creedmoor and .260 Remington that are used in training set, with much heavier bullet and wider effective range, so this set could test the performance of the model on a dataset that is less similar with training set.
No. of Ammos in table 3.3 refers to the number of data in each set. Each ammunition data consists of one shoot, with nine measures of velocity at different distance from the muzzle.

The data set was chosen from the book Ammo & Ballistics 6: For Hunters, Shooters, and Collectors by Forker[14], which was published in the United States and uses the US system of measurement units. Because of the country's relatively less strict gun laws, there are more people who own and use firearms in the United States.There are a number of federal laws in the US that control access to firearms and these laws regulate how guns, ammo, and related equipment are produced, exchanged, stored, transferred, kept in records, transported, and destroyed[4]. Out of the fifty states, forty-four have a clause in their state constitutions that protects the right to keep and bear arms, comparable to the Second Amendment of the US Constitution. Due to all of these factors, the major part of ballistics and guns data is compiled in the United States and recorded in the US unit system.
In my coding and this report, I will utilise the US unit system in my programming rather than the more often used International System of Units (e.g. gramme, metre).

3.3 Equation of Motion

Equations of motion are formulas that explain how a physical system operates in terms of how its movement changes over time, thus the behaviour of a bullet satisfies an equation of motion. The generic vector differential equation of motion for constant projectile mass is given by Newton's second law as:

$$m\frac{d\overrightarrow{V}}{dt} = \sum \overrightarrow{F} + m\overrightarrow{g} + m\overrightarrow{\Lambda} \tag{3.2}$$

Where m represents the projectile mass, $\overrightarrow{V}$ represents the vector velocity, $\sum \overrightarrow{F}$ represents the vector sum of all aerodynamic forces the projectile experiencing, $\overrightarrow{g}$ the acceleration due to gravity and $\overrightarrow{\Lambda}$ the Coriolis acceleration due to the earth's rotation.
Coriolis acceleration can significant affect projectiles with large weight and long range travel distance, such as launching a worldwide missile, however, in comparison to the acceleration caused by gravity, the Coriolis acceleration is minimal for cartridges that are light in weight and have a short shooting range. An assumption of ignoring the Coriolis acceleration is made in this dissertation.
The model using equation of Motion will be discussed in detail in Chapter 4.

3.4 Regression

There are many algorithms in machine learning that could model a trajectory of projectile. However, many of the methods are "black boxes". A model is referred to as a "black box" if it is so complex that it is difficult for humans to understand it. Petch *et al.*[36] stated some of the limitations of lack of interpretability machine learning models in Cardiology, and some of them are still applicable in engineering. The reliability of predictive models can be diminished, and the model's ability of predicting can be restricted if the trained model is not

expressible. Also, in order to track the trajectory, improve understanding of the behaviour of the projectile, and apply this research finding to engineering design, using regression method which can give a clear trajectory that describes the behaviour of the projectile is a better choice, comparing to other "black box" machine learning algorithm.

3.4.1 Symbolic Regression

As discussed in the Chapter 2, genetic programming algorithm was chosen to solve symbolic regression problem. The tool that I choose for genetic programming is DEAP. DEAP (Distributed Evolutionary Algorithms in Python)[39] is designed to render algorithms explicit and data structures accessible. DEAP is open sourced, well documented[1] and already have a few completed cases and researches based on DEAP to reference online comparing with other open source projects.

3.5 Evaluation and testing

The aim of this dissertation is to find a machine learning based solution to simulate the movement of a bullet. By using the method of symbolic regression, it expects an interpretable function made up of given mathematical operations. This function can help with understanding the trajectories of different types of bullets, predict the movement of bullets, and could even possibly help to simulate the trajectories of bullets not in the database, reduces financial cost, material wastes and time consuming. The results will be examined using both test sets that are similar to the training set (cartridge size close to the 6.5mm) and test sets which are less similar (big cartridge with heavier bullet) in order to assess the accuracy and generality of the model developed, although for the test set that are far away from the training set, the model is less likely to have a good performance on it.
There are a number of commonly used accuracy measurement methods, which can be divided into four types: scale-dependent measures, percentage errors based measures, relative errors based measures, and relative measures[19]. This project does not include comparisons across multiple scales, thus scale-dependent measurements should sufficient. The commonly used methods are Root Mean Squared Error(RMSE), Mean Absolute Error(MAE) and Mean Absolute Percentage Error(MAPE). The error value of for predicted values $\hat{y}_t$ for times t of a regression's dependent variable y_t is calculated as:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}. \tag{3.3}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \tag{3.4}$$

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{\hat{y}_t - y_t}{y_t} \right| \tag{3.5}$$

Comparing with MAE and MAPE, RMSE is more sensitive to outlier value, and RMSE is in the same unit of measurement as the variable, so RMSE is selected for testing in this project.

3.6 Ethical, Professional and Legal Issues

This dissertation does not include an ethical review. The datasets used in this project are either publicly accessible or simulated. This research is constrained by a lack of quantity and variety of data, as conducting shooting tests is impractical for me. Any further ammunition-related work must adhere to local firearms regulations.

Chapter 4

Planning

4.1 Data Normalisation

It is difficult for DEAP to add constant terms to the function and provides a reliable prediction of large numbers. Moreover, comparing to the features such as distance and velocity, the Boolean variables `boattail`, `roundnose` and `cannelure` have a small value of either 0 or 1, and these small values are usually ignored by the algorithm when constructing estimating function, as the values are tiny comparing to the target value, which is the velocity. As a result, it is necessary for the dataset to be normalised. The methods that will be attempted in this projects are `StandardScaler`, `MinMaxScaler` and `MaxAbsScaler` from the `scikit-learn` Python library[8][35].

- `MinMaxScaler`: A transformation that changes the data to fit inside a certain range. The transformation is given by:

$$X_{MinMax} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

- `MaxAbsScaler`: Each feature is scaled by the maximum value. The transformation is given by:

$$X_{MaxAbs} = \frac{x}{|x_{max}|} \tag{4.2}$$

- `StandardScaler`: Each feature is standardised using mean and standard deviation. The transform is given by:

$$X_{standard} = \frac{x - x_{mean}}{s.d.} \tag{4.3}$$

- Initial velocity: This is a custom method for normalisation. For each bullet, the velocity is scaled by divided by the initial velocity, and other features are scaled using a method from the methods mentioned above. In this experiment I choose to use `MaxAbsScaler`

to scale features other than velocity, as it is the method with the lowest RMSE among first three(result shows in next paragraph).

In addition to the standard ways to normalisation, I will test a custom method, which is stated as "Initial velocity" above. All the method mentioned above are applied on the whole dataset instead of normalising data for individual bullet. The method I purpose for this project is normalise the velocity by dividing the velocity by the initial velocity of each bullet. This approach has its benefits and drawbacks. By applying this normalisation method, the initial velocity of each bullet will be change to 1, this allows the symbolic regression to learn the feature of "velocity at $d = 0$ should equal to initial velocity". On the other side, applying different scale to each bullet leads to an inconsistency in measurement. A comparison of different methods of normalisation is taken. For each method, the same parameters are set for the genetic programming, and the algorithm is run multiple time(twenty times in this experiment) on each normalisation method and taken a mean value for a fair comparison:

	RMSE
No normalisation	546.238
`MinMaxScaler`	522.769
`MaxAbsScaler`	421.591
`StandardScaler`	430.042
Initial velocity	550.852

Table 4.1: RMSE of model trained with different normalisation methods

In this chart I compare different methods and their performance using RMSE measurement. RMSE row shows the Root-mean-square error, which is the difference of prediction value compares with the target velocity at different distance, this value could represent the accuracy of the relative method. The RMSE is calculated in the scale of original data, as for each normalisation method, the dataset is transformed in different scales, it is necessary to transform the data to the original unit, otherwise it is worthless to compare values in scale. In this experiment, the techniques such as add penalty to model not starting from correct initial velocity are not yet applied, which is also a reason of why the model does not perform well. From the result of experiment, it is shown that comparing to other three method, the `MaxAbsScalar` has the best performance among all four methods. `StandardScaler` also has a good overall performance, however, there is problem with using `StandardScaler` for normalisation. Standard scaler method assumes that dataset fits in a standard distribution, which does not discribe the distribution of current dataset. One thing that I notice during the experiment is the algorithm is very likely to give an estimating function of $f = -D$, as the value of distance increases, the value of velocity falls, and both distance and velocity shift from being far from the mean to the mean and then away from the mean again. The model learns the correlation between distance and velocity and their mean, but this is not the proper pattern to learn for this project. In this dissertation I will use `MaxAbsScaler` as

the evaluation method as it shows a good performance in this experiment.

4.2 Theoretical approach

In this session I will use the Equation of Motion approach to calculate the movement of bullet, the equation 3.2 reduces to:

$$m\frac{d\overrightarrow{V}}{dt} = \sum \overrightarrow{F} + m\overrightarrow{g} \tag{4.4}$$

The forces that influence the trajectory of a bullet include the drag force, the lift force and the Magnus force, which is the force acting on the spinning object when it moves through a fluid. A few assumptions have been made: With the assumption that the bullet is shot horizontally and with a minor elevation angle, the Coriolis acceleration caused by the earth's rotation is neglected in this equation, as well as the lift force on the projectile. Since the Magnus force is small in comparison to either the aerodynamic drag force or the force caused by wind drift, it is likewise ignored in this situation.
As a result, in this project, the $\sum \overrightarrow{F}$ may be divided into two types: aerodynamic drag force and the horizontal force that causes a bullet to be pushed off path by a crosswind. The drag force has magnitude of $(\frac{1}{2})\rho V^2 S C_D$ and direction in $-\overrightarrow{V}$. Hence, the vector form is provided by:

$$\overrightarrow{F_D} = -\frac{1}{2}\rho S C_D V \overrightarrow{V} \tag{4.5}$$

Where S represent the projectile reference area. C_D refers to the drag coefficient of bullet, in this project G1 drag model is used, and the value for G1 drag coefficient is obtained from McCoy[33]:

Mach number	C_{D1}
0	0.263
0.5	0.203
0.6	0.203
0.7	0.217
0.8	0.260
0.9	0.342
0.95	0.408
1.0	0.435
1.05	0.543
1.1	0.588
1.2	0.639
1.3	0.659
1.4	0.663
1.5	0.657
1.6	0.647
1.8	0.621
2.0	0.593
2.2	0.569
2.5	0.540
3.0	0.513
3.5	0.504

Table 4.2: drag coefficient of G1 drag coefficient, data from McCoy[33]

I observed from table 4.2 that C_{D1} is given only for a few velocity number, in order to estimate the drag coefficient between each Mach number interval, I assume the relationship of velocity and drag coefficient is linear inside each interval. The pseudocode below demonstrates a function I use to calculate the estimated drag coefficient:

Algorithm 1 Drag Coefficient estimation

Input Reference table {Mach Number, C_D}=(a_n, c_n), current velocity v
When $a_i < v < a_{i+1}$
return $c_d = c_i + \frac{c_{i+1}-c_i}{a_{i+1}-a_i} * (v - a_i)$
Break

In the data set, there is a crosswind of 10mph involves. Crosswind will add a force to the bullet on the side, and will affect the movement of bullet on the horizontal xy-plane. The figure 4.1 shows the xy-phase and direction of forces acting on the bullet.

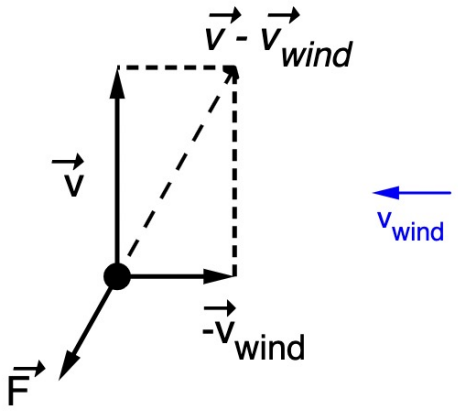


Figure 4.1: An illustration of direction of force acting on a flying bullet

The drag force act against the moving direction of the bullet, and the direction is affected by the crosswind. When the impact of the crosswind is taken into account, the force on the direction of wind leads to[31]:

$$F_x = -\frac{C_D}{2}\rho S v v_{wind} \tag{4.6}$$

The acceleration on each direction can be therefore deduced:

$$\frac{d\overrightarrow{V}}{dt} = \dot{V}_x \overrightarrow{I} + \dot{V}_y \overrightarrow{J} + \dot{V}_z \overrightarrow{K} \tag{4.7}$$

$$\dot{V}_x = -\frac{C_D}{2m}\rho S v v_{wind} \tag{4.8}$$

$$\dot{V}_y = -\frac{C_D}{2m}\rho S v v_y - g \tag{4.9}$$

$$\dot{V}_z = -\frac{C_D}{2m}\rho S v v_z \tag{4.10}$$

Noted in the equations 4.8, 4.9 and 4.10 the projectile has been symplified to a cylinder and the ballistic coefficient B_c has been written as $\frac{m}{d^2}$, where d represents the diameter of the projectile. The acceleration should convert to:

$$\dot{V}_x = -\frac{\rho \pi C_D}{8B_c} v v_{wind} \tag{4.11}$$

$$\dot{V_y} = -\frac{\rho\pi C_D}{8B_c} v v_y - g \tag{4.12}$$

$$\dot{V_z} = -\frac{\rho\pi C_D}{8B_c} v v_z \tag{4.13}$$

I choose to use tangent line approximation to estimate the value of equation 4.4, the algorithm is shown below:

Algorithm 2 Tangent Line Approximation

Input $v \leftarrow$ initial velocity, $f'(x) \leftarrow$ function dv/dt, $D \leftarrow$ distance, $a \leftarrow$ step size, initial distance $d \leftarrow 0$, initial time $t \leftarrow 0$
while $d < D$ **do**
  $f(t+a) = f(t) + af'(t)$
  $t \leftarrow t + a$
  $d \leftarrow d + \frac{f(t+a)+f(t)}{2} * a$
  Save velocity $v \leftarrow f(t+a)$ at distance d
  Break
return velocity v at relevant distance d

Inaccuracy of this method:

- In this project, in order to estimate the affect of crosswind on the bullet, the shape of bullet is simplified into a cylinder, so the cross sectional area can be approximate using the base area of cylinder, which is equivalent to πr^2. But in practice, the cross section of projectile is not constant from the tail to tip and the area cannot be easily represented by using this method.
- The drag coefficient curve is steep at certain intervals, and the change within an interval could be inconsistent. The coefficient of drag calculated based on the assumption of a linear relationship within each speed interval may differ substantially from the actual value.
- Note that in the algorithm, an average of start and end velocity at each step was used for estimation of distance. This may leads to an underestimation of travel distance, and the velocity at each distance may less than the analytical solution. The use of lower step sizes is a potential answer for this concern.

4.3 Symbolic Regression

Symbolic regression method was chosen for this project, the reason and benefits of using symbolic regression are explained in previous chapter. The Python library I picked to achieve symbolic regression is DEAP. An example using DEAP to solve symbolic regression problem looks like this:

```python
def protectedDiv(left, right):
    try:
        return left / right
    except ZeroDivisionError:
        return 1

pset = gp.PrimitiveSet("MAIN", 1)
pset.addPrimitive(operator.add, 2)
pset.addPrimitive(operator.sub, 2)
pset.addPrimitive(operator.mul, 2)
pset.addPrimitive(protectedDiv, 2)
pset.addPrimitive(operator.neg, 1)
pset.addPrimitive(math.cos, 1)
pset.addPrimitive(math.sin, 1)
pset.addEphemeralConstant("rand101", lambda: random.randint(-1,1))
```

Figure 4.2: Creating primitive set and add operators in DEAP[3]

```python
pset.renameArguments(ARG0='x')
creator.create("FitnessMin", base.Fitness, weights=(-1.0,))
creator.create("Individual", gp.PrimitiveTree, fitness=creator.FitnessMin)
```

```python
toolbox = base.Toolbox()
toolbox.register("expr", gp.genHalfAndHalf, pset=pset, min_=1, max_=2)
toolbox.register("individual", tools.initIterate, creator.Individual, toolbox.expr)
toolbox.register("population", tools.initRepeat, list, toolbox.individual)
toolbox.register("compile", gp.compile, pset=pset)

def targetfunc(x):
    return x**4 + x**3 + x**2 + x

def evalSymbReg(individual, points):
    # Transform the tree expression in a callable function
    func = toolbox.compile(expr=individual)
    # Evaluate the mean squared error between the expression
    sqerrors = ((func(x) - targetfunc(x))**2 for x in points)
    return math.fsum(sqerrors) / len(points),

toolbox.register("evaluate", evalSymbReg, points=[x for x in range(-10,10)])
toolbox.register("select", tools.selTournament, tournsize=3)
toolbox.register("mate", gp.cxOnePoint)
toolbox.register("expr_mut", gp.genFull, min_=0, max_=2)
toolbox.register("mutate", gp.mutUniform, expr=toolbox.expr_mut, pset=pset)

toolbox.decorate("mate", gp.staticLimit(key=operator.attrgetter("height"), max_value=17))
toolbox.decorate("mutate", gp.staticLimit(key=operator.attrgetter("height"), max_value=17))
def main():
    stats_fit = tools.Statistics(lambda ind: ind.fitness.values)
    stats_size = tools.Statistics(len)
    mstats = tools.MultiStatistics(fitness=stats_fit, size=stats_size)
    mstats.register("avg", numpy.mean)
    mstats.register("std", numpy.std)
    mstats.register("min", numpy.min)
    mstats.register("max", numpy.max)
    pop = toolbox.population(n=300)
    hof = tools.HallOfFame(2)
    pop, log = algorithms.eaSimple(pop, toolbox, 0.5, 0.2, 40, stats=mstats,
                                   halloffame=hof, verbose=True)
    return pop, log, hof
```

Figure 4.3: Creator and Toolbox in DEAP[3]

The `targetfunc` in figure 4.3 is the target function. The model DEAP returns has the format of:

```
print("\nBest Hof:\n%s"%hof[1])
```

```
Best Hof:
add(add(add(mul(mul(x, x), x), x), mul(x, x)), mul(x, mul(x, mul(x, x))))
```

Figure 4.4: Example of function provided by DEAP

In order to train the model, a reasonable selected and preprocessed variables are necessary. One characteristic of DEAP is that DEAP performs bad on large values. Some of the common operations that are used by DEAP to form the functions have a restricted range, for example, the outcomes from $sin(x)$ and $cos(x)$ can only landed between $(-1, 1)$. Moreover, DEAP lack the ability of adding constant term in the equation, the only way of producing a constant term is divide the varible by itself. Use the example in figure 4.3 as example, when all other features of the algorithm stays the same:

Table 4.3: Example of using DEAP for symbolic regression

Target function	at x=1	Symbolic regression result	at x=1
$x^4 + x^3 + x^2 + x$	4	$x(x^3 + x^2 + x + 1)$	4
$x^4 + x^3 + x^2 + x + 1000$	1004	$x(x^2(x + \frac{x}{x}) + 16x + 2\frac{x}{x})$	20

The target of prediction in this project is the velocity of bullet, and the velocity value is usually in thousands. After attempting training without any preprocess, the outcome value usually far away from the real value. One of the solution is scaling the data. There are many aproaches to scale a dataset, some examples of scaling are `StandardScaler`,`MinMaxScaler` and `RobustScaler`. Different scaling methods could possibly affect the accuracy of machine learning model, which has been discussed earlier in this chapter. Ahsan *et al.*(2021)[5] evaluated different algorithms for machine learning using various data scaling techniques, which confirmed the effect of scaling method on the performance. Ahsan *et al.*(2021) did not include any investigation on symbolic regression, so I did not take their conclusion as a solid reference. The choice of normalisation method has been discussed in session 4.1
As mentioned in Chapter 2, the ballistic provided by manufacturer is not always accurate. In order to avoid the misleading ballistic coefficient value, I choose to train the model without the involving of ballistic coefficient value. The features and target for the symbolic regression model are shown below:

- `D`: refers to the distance of bullet measured from muzzle. In the dataset given, the velocity was measured at 0, 100, 200, 300, 400, 500, 600, 800, 1000 yards from the muzzle.
- `Weight`: the weight of bullet measured in grains.
- `boattail`: Boolean variable. The design of bullet with a boat tail is assigned value 1, otherwise 0.

- `roundtip`: Boolean variable. It has been studied that the shape of bullet tip affects the aerodynamic drag forces on the bullet. A bullet with round bullet tip is assigned value 1, otherwise 0.
- `cannelure`: Boolean variable. A cannelure is a groove around the ammunition. A cannelure on the bullet can prevent the bullet from being pressed deeper into the cartridge as a result of the gun's inertial action, which is a part of interior ballistics. This variable should have less impact on the trajectory other than other variables, while it is interesting to observe if the result is relevant with this variable.
- `initialVelocity`: The initial velocity of bullet at the muzzle. As this is a predicting model of exterior ballistics, the initial velocity need to be given for estimation.
- `Velocity`: Target of this model. This data is provided in unit of feet per second.

As the task of this project is to predict a dynamic model, the prediction result needs to satisfies a few condition to make the function "seems like" a description of projectile movement in real life:

- **Velocity at $d = 0$:** One possible issue of symbolic regression is the velocity curve does not start from the initial velocity.

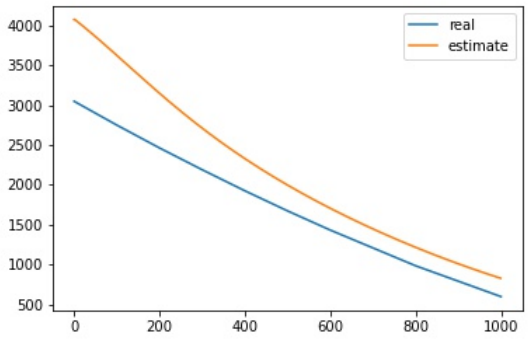


Figure 4.5: Example of predicting model not start from initial velocity

A typical prediction model needs to begin with the provided beginning velocity. The model appears unnatural and is worthless as a reference if the initial velocity is incorrect. I put a penalty to the model when the model picks the incorrect initial velocity for this issue.

- **Fluctuating velocity:** The velocity of bullet needs to follow the physical law. In the real life, the velocity of a projectile can only reduce with time when it does not experience any external force since it is subject to the law of conservation of energy.

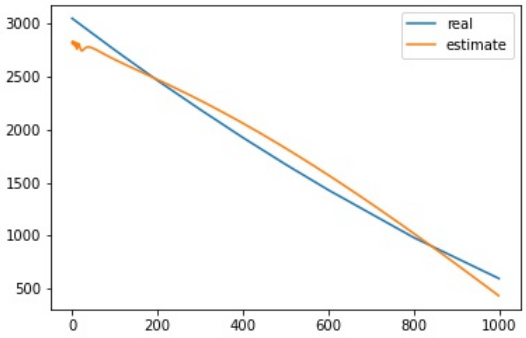


Figure 4.6: Example of velocity not decline over time

At the left side of figure 4.6, the velocity of prediction model shows an act of fluctuation, which is unrealistic to happen in real-life scenario. After the function is generated in the model, I will randomly select distance values with small gap and compute the velocity. For distance $d_1 > d_2$, velocity should act as $v_{d_1} > v_{d_2}$, otherwise a penalty will apply to the function.

- **Change of acceleration:**

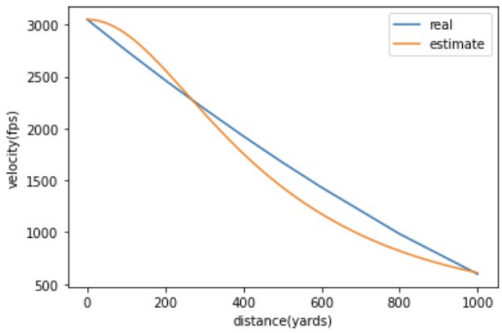


Figure 4.7: Example of acceleration not decrease steadily

Figure 4.7 shows an example of projectile acceleration not constantly decrease. The estimate curve shows a decreasing trend of "flat-steep-flat", indicating that the bullet's rate of acceleration is not monotonously decreasing. This abnormal trend could be detected by computing the derivative of acceleration(*i.e.* second order derivative of velocity). The derivative of acceleration should always be negative to satisfy the behavior of a projectile. An analytical differentiation process can be achieved through `SymPy`, by simply using `sympy.diff()`. However, this takes long processing time. Instead of using analytic derivative, a numerical derivative estimation could be applied:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{4.14}$$

where h is a small number.

For second order derivatives such as acceleration, apply the equation 4.14 a second time:

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \tag{4.15}$$

In this project, the function $f(x)$ usually refers to the velocity of projectile, and h refers to a small time gap and is set to be 0.0001 second.

Algorithm 3 is a pseudocode showing the rule of adding penalty in the evaluation process:

Algorithm 3 Evaluation process

Input training set D_{train} includes distance d, ballistic coefficient B_c, initial velocity IV *etc.*, velocity estimation function $f(X)$
for each set of data X in D_{train} **do** measurement of error $error = 0$
  if $d = 0$ and $f(X) = IV$ **then**
    No penalty
  else if $d = 0$ and $f(X) \neq IV$ **then**
    $error += RMSE + penalty$
  else
    if $\frac{dv}{dt} > 0$ **then**
      $error \leftarrow error + RMSE + penalty$
    else if $\frac{d^2v}{dt^2} > 0$ **then**
      $error \leftarrow error + RMSE + penalty$
    else
      $error \leftarrow error + RMSE$
return $error$

The penalty value added in algorithm 3 is large comparing with the RMSE values. In practice, I set the penalty value to be 10^8, while the RMSE values are usually around one hundred.

4.3.1 Genetic programming process

Genetic programming is an algorithm that can evolve an inadequate programme until it satisfies the standard. It operate on the population of programmes through methods similar to those seen in natural genetic processes. There are several types of genetic programming, this project uses the tree-based genetic programming, which describes a programme with tree structure and iteratively evaluates through the tree to produce the final expression. Figure 4.8 shows a tree-based expression of function $x * y + (-z)$.

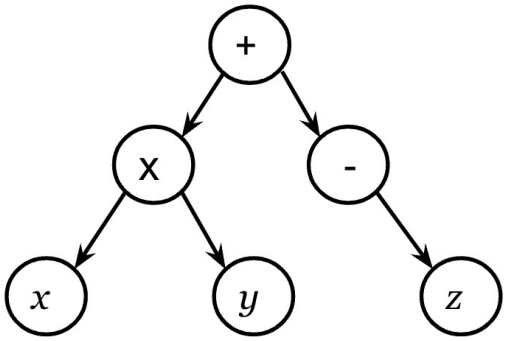


Figure 4.8: An example of tree-based expression of $x * y + (-z)$

The genetic programming proceeds as follows:

1. A primitive set with operators is generated. Any operations that will be used later in the functions should be added to the primitive set. The operations I choose for this stage are `add(a,b)`, `sub(a,b)`, `mul(a,b)`, `div(a,b)`, `scala2(a)`$(= 2a)$, `add2(a,b)`$(= a+2b)$, `mul2(a,b)`$(= a * 2b)$ and `div2(a)`$(= \frac{a}{2})$. `scala2(a)`, `add2(a,b)`, `mul2(a,b)`, `div2(a)` are a combination of multiple operations, this might decrease the overall number of operators in the function and speed up the training process, as there are fewer generators in each individuals that need to operate on. It is also worth to mention that there is an if statement inside `div(a,b)` function, which protects the function from returning a zero division error and returns a value of one if divisor equals to zero.

2. Register some parameters to the genetic programming. This refers to `toolbox` in the example in figure 4.3. This includes specifying how the tree is generated, restricting the depth of the tree, establishing a static limit for some certain measurements defined by Koza[23]. Last but not least, the measurement of error for genetic programming generated function should be stated and added to the toolbox of the algorithm. The error measurement function should have an input of function variables and their target value, and returns a value that represents the error of the function.

3. Launching the evolutionary process. In this step, the population of genetic programming is specified, which is the size of possible solutions that are considered in this generation. Any parameters pertaining to evolution, such as the probability of mating and mutation, should be entered at this stage in order to prepare for iteration. An example could be find in the `eaSimple` function in figure 4.3.
   It is worth noted that the choice of population size is important. If the population size is too small, the model may slip into local optimization and struggle to produce well-performed solutions, on the other hand, a large population can make the training process computationally expensive[28].

4. An initial population is generated at random, following by a specified number n of generations of the algorithm:

   (a) Each individual in the population are evaluated. The top k individuals with the best performance are selected, while the others with lower performance are eliminated.

   (b) The selected individuals then crossover at a certain probability. New individuals are produced and added to the population.

   (c) Each individual will mutate at certain probability. Mutation could introduce diversity into the model, and avoid the model fall into local optima by preventing only similar chromosomes(features) selected in the population.

5. The operations in step 4 are repeated for n generations(*i.e.* iterations of genetic programming), and the best performance functions will be returned.

Here is a flow chart that demonstrates this process:

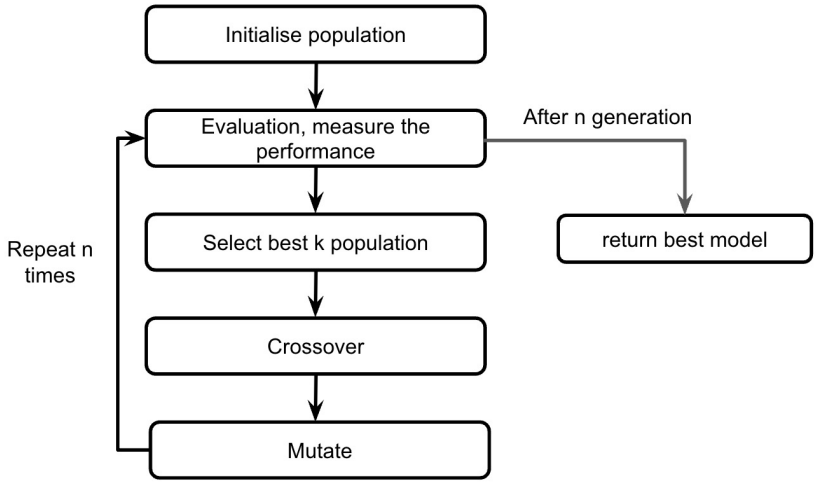


Figure 4.9: Flowchart of genetic programming training process

4.3.2 SymPy

The Python library `SymPy` is used for converting functions generated by DEAP to readable results. `SymPy` is a open source computer algebra system, it is capable to handle a series of tasks from basic arithmetic to solving equations and calculus[43][21].
`SymPy` can convert DEAP results to a easily readable format or format that suits LaTeX. Here are the steps of transforming DEAP results to other formats:

1. Convert some mathematical operations and customised operations into `SymPy` readable format:

DEAP operator	Maths expression	expression in `SymPy`
add(a,b)	$a + b$	Add(a,b)
sub(a,b)	$a - b$	Add(a, Mul(-1,b))
mul(a,b)	$a * b$	Mul(a,b)
div(a,b)	a/b	Mul(a,Pow(b,-1))
scala2(a)	$2a$	Mul(2,a)
add2(a,b)	$a + 2b$	Add(a,Mul(2,b))
mul2(a,b)	$a * 2b$	Mul(a,Mul(2,b))
div2(a)	$a/2$	Mul(1/2,a)

Table 4.4: Operators in DEAP and `SymPy`

   Noted that last four operations are used to reduce the need of populations in the genetic programming for symbolic regression.

2. Convert the expression into string.

3. Apply `sympy.simplify()` to the function, this will give the function a simplest form of expression string and print the result in a easily readable format. This is useful especially when the population of genetic programming is set up too high, and the output function contains unnecessary terms(*e.g.* $x * (x * x^{-1})$ instead of x).

Here is an example of converting DEAP result process:

1. DEAP output: `mul(mul(mul(add(x, protectedDiv(add(x, add(protectedDiv(add(x, protectedDiv(x, x)), x), sin(sub(x, x)))), x)), x), x), x)`

2. SymPy expression in string: `'Mul(Mul(Mul(Add(x,Mul(Add(x,Mul(Add(x,Mul(x, Pow(x, -1))), Pow(x, -1))), Pow(x, -1))),x),x),x)'`
   This can be written as: $(x + (x + x * x^{-1}) * x^{-1}) * x * x * x$

3. Simplify the function through `SymPy` and return the LaTeX format: $x\left(x^3 + x^2 + x + 1\right)$

4.4 Hybrid Model

Instead of training the symbolic regression model only based on the dataset, there is another possible approach of a combination of both equation of motion and symbolic regression. The $\sum \overrightarrow{F}$ term in equation 4.4 is made up of several forces, and it is not accurate or comprehensive to estimate the force using aerodynamic drag force only. Gravity undoubtedly affects how a projectile moves, therefore the term $\overrightarrow{g}$ may be left in the equation 4.4 and I choose to use

symbolic regression to estimate the aerodynamic forces applied on the projectile. The target of this session should become:

$$m\frac{d\overrightarrow{V}}{dt} = func(X) + \overrightarrow{F} + m\overrightarrow{g} \tag{4.16}$$

Where *func( )* represents the aim of symbolic regression, X represents variables that may use in the equation, and $\overrightarrow{F}$ represent the drag force that is mentioned in the equation of motion session 4.2, so the *func( )* is used for filling in the gap between theoretical approach and the real value.
The setup and procedure of building up this hybrid model is similar to the symbolic regression model, but the variables used and target value is different. The estimating function in this task, as mentioned in the previous paragraph, attempts to narrow the gap between the theoretical method and the actual value, so the target variable of this symbolic regression model is the difference of theoretical method velocity and the velocity from the dataset and an evaluation method(RMSE in this project) is taken to test the performance of the model.

Just same as the symbolic regression model, the hybrid model needs to satisfy the law of motion to ensure the model can be applied to real life. In this case, the method of measurement is simpler: As the right hand side of equation 4.16 $func(X) + \overrightarrow{F} + m\overrightarrow{g}$ describes the acceleration of the projectile, the acceleration needs to be in the direction of against the movement of bullet, The estimation of first and second order derivative of $m\overrightarrow{V}$ should satisfy the requirement of conservation of energy, which means, the . first and second order derivative of $func(X) + \overrightarrow{F} + m\overrightarrow{g}$ should be both negative.
Variables and target that will be used in this method are stated as below, most of the variables have already been explained in the symbolic regression session:

- `D`: Distance measured from muzzle.
- `Weight`: weight of bullet in grains.
- `boattail, roundtip, cannelure`: Boolean variable related to the shape of bullet.
- `initialVelocity`: Initial velocity of the bullet at muzzle.
- `Veom`: Velocity at current position computed by the equations of motion.
- `dragCoefficient`: It has been demonstrated that the drag coefficient can influence the aerodynamic drag force and thereby the bullet velocity[25][45]. Therefore I add an variable of drag coefficient, which could be estimated using the drag coefficient estimation function (algorithm 1) and current scalar velocity of the bullet.
- `V_gap`: target of this function, the velocity difference of equation of motion and the target value.

4.4.1 Limitation of the project

At first, the plan of this project is to use symbolic regression to estimate forces that the projectile experiencing in each direction, and use tangent line approximation to compute the estimate velocity in 3 degrees of freedom. Upon attempting, I discovered that this approach is not realistically achievable: The tangent line approximation needs to be iterated multiple times, in each iteration each function is calculated three times in different direction, and the iterations are computed at a small time gap step to achieve greater precision during the evaluation stage. This requires a considerable amount of calculations. Although this project is not a time-sensitive task and training time should not be considered as a main factor in this project. This calculation is hard to achieve with the performance of my personal computer. Instead, I choose the current approach, which is to improve the accuracy of equation of motion with symbolic regression. Although the current method has slightly deviated from the original intention of this session, which is using symbolic regression technique to give a precise prediction of $\frac{d\overrightarrow{V}}{dt}$ in all different directions, with my existing equipment, the desired goal could not be attained.

4.5 Internal ballistics modelling

The internal ballistics modelling task involves in this project is the relation of gunpowder loaded and initial velocity of bullet. However, an experimental data including different volume of powder loaded and and measurement of initial velocity is hard to find. I use simulated data from Gordons Reloading Tool[2], which can produce data through mathematical calculation of internal ballistics by selecting cartridge, projectile and powder loaded volume. Comparing to the dataset for external ballistic models that are discussed earlier in this dissertation, which only has one set of data for each cartridge and bullet combination, this time, a range of data with different gun powder volume are used for training. In this experiment, not only performance of symbolic regression on internal ballistics, but also the effect of broadness of data on symbolic regression could be tested. For each cartridge and projectile, a function that describes the relation of powder volume and initial velocity of projectile is trained base on following features via symbolic regression:

- **Cartridge type:** These are not variables included in the estimating function. Although GRT database includes detailed measure of the cartridges and their chamber, considering it requires a large amount data of measurements in order to investigate how initial velocity number affected by cartridge shape, and all data is manually entered, it is not realistic to study this topic in this project. Instead of computing a general function that suits all cartridges, I choose to train a estimating function for each cartridge individually.
- **bullet measurement:** A series of bullet measurements such as diameter, length, weight, tail type and ballistic coefficient are contained in the dataset. The unit of all measurements are in inches.

- **powder loaded:** Propellant in the chamber of cartridge burns and generates large amount of gas and pushes the projectile out of the barrel. In the safe load range, for the same propellant and a typical, no aberrant shot, more gunpowder loaded usually means faster initial speed. The powder loaded values are measured in grains.
- **Initial velocity:** The initial velocity in this session refers to the velocity of bullet at the muzzle. While there are many factors that might affect initial velocity, in this project, I only focus on the impact of gunpowder loaded weight. The initial velocity is measured in feet per second.

For this task, I achieve symbolic regression using the DEAP framework. Symbolic regression method and DEAP has been discussed in chapter 2, 3 and 4. The data is manually entered and generated by GRT for each bullet, providing the initial velocity from a powder load of 30 grains to the maximum load, which is often around 40 grains. The full size dataset contains 10 6.5mm Creedmoor ammunitions and their measures, powder loaded and initial velocity, and the test dataset contains 2 ammunitions of same type.
In this session I also test the performance of symbolic regression model on different size of datasets. I reduce the size of dataset by randomly removing some of the rows. The four datasets I choose to test the model on are 100%(full dataset), 70%, 40%, 10% and their results are discussed in chapter 5.

Chapter 5

Result and Discussion

5.1 Models included in this project

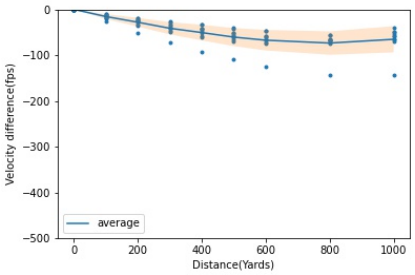
Several models are trained and their performances is tested:

- Theortic approach using equation of motion with ballistic coefficient
- Theortic approach using equation of motion with mass and diameter
- Symbolic regression on velocity bases on ballistic model
- Symbolic regression on velocity bases on ballistic model + penalty at evaluation
- Hybrid model: Symbolic regression on $\frac{dv}{dt}$
- Symbolic regression on internal ballistic modelling

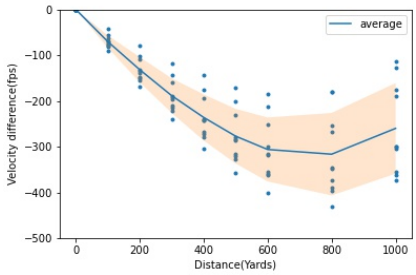
In this session, I will discuss and compare the performance of some of the models above. I choose to use RMSE to analyse the error of each model.

5.2 Comparison of equation of motion with and without ballistic coefficient

Figure 5.1 and figure 5.2 shows the results of theoretical approach using equation of motion on the training set, which includes 6.5mm Creedmoor and .260 Remington, two ammunitions that have similar size and performance. Two types of equation of motion are used: Figure 5.1a demonstrates the difference of equation of motion using ballistic coefficient(equation 4.11, 4.12 and 4.13) and training set by different distance from the muzzle. Figure 5.1b demonstrates the difference of equation using mass and diameter to estimate the ballistic coefficient(equation 4.8, 4.9 and 4.10) and the training set by different distance from the muzzle. The shade in figure 5.1 has a width of two standard deviation.
Figure 5.2 plots the average of two models in the same figure for easier comparison.



(a) Error using ballistic coefficient estimation



(b) Error using mass-diameter estimation

Figure 5.1: Error using equation of motion with different methods on train set

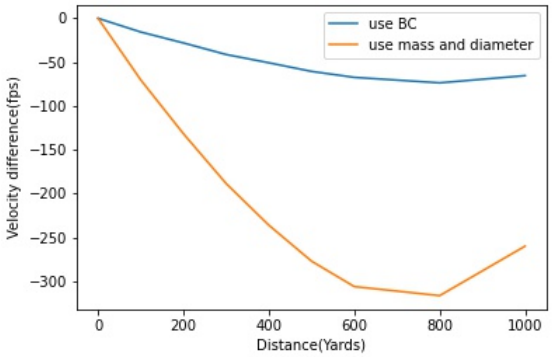


Figure 5.2: Average error of equation use ballistic coefficient and mass-diameter

Distance(Yard)	Ballistic Coefficient	Mass and diameter
0	0	0
100	-15.612	-70.350
200	-28.105	-131.807
300	-41.297	-188.646
400	-50.733	-236.180
500	-60.588	-277.235
600	-67.386	-306.254
800	-73.644	-316.579
1000	-65.427	-260.114

Table 5.1: Comparison on average error of two equation of motion model

From figure 5.1a and figure 5.1b it can be seen that both ballistic coefficient and mass-diameter does not give an accurate velocity estimation. Both of the model underestimate the velocity.

It could be seen from figure 5.1 that mass-diameter model has a larger standard deviation comparing to the ballistic coefficient model, the reason of this is using mass and diameter to estimate ballistic coefficient ignore the impact of bullet shape, material and design, and for all bullets in the training set, the same diameter value is applied as the bullets has same calibre. This causes the estimation values spread out and make the result less reliable.
Figure 5.2 compares the mean value of two models in the same plot. The figure shows that the error of mass-diameter model is much greater than ballistic coefficient model. The trustworthiness of mass-diameter model is questionable, but the performance of ballistic coefficient model is acceptable. In this chapter later, I will develop the equation of motion with ballistic coefficient by combining it with symbolic regression and try to further improve the performance.

5.3 Symbolic regression on projectile movement

First I attempted the model without penalty added for the genetic programming evaluation process. If only consider the RMSE value and the deviation from real value is acceptable. However, the estimating functions produced by the genetic programming often do not satisfy the requirement of this project, which is to simulate a real-world movement that follows the law of motion. The majority of functions have the problems of not following the initial velocity(*i.e.* return a different value from the initial velocity at distance of 0), or the velocity does not decrease as the bullet travels further. As discussed in chapter 4, different levels of penalty was added to the model in order to achieve a reasonable result.
At first each penalty in evaluation was set to a same value. A phenomenon of model converging to a local optimum was observed, and in order to solve this issue, I took a few actions: After a few experiments, it seems that assigning different level of penalty value can improve the performance of model, and reduce the chance of machine learning model falling in to local optimum to avoid being excessively penalised. Another measurement I choose is to increase the mutation rate and population number of genetic programming. A higher mutation rate brings more diversity to the population(Cobb, 1990)[11], and a larger population number could also keep the sub-optimal solutions to ensure the model searches in a wider range of possible optimisation solutions.
Another observation is that the symbolic regression model occasionally returns a function that is only related to the variables "distance" and "initial velocity", while ignoring variables that have been demonstrated to be theoretically relevant to velocity, such as "ballistic coefficient" and "weight." A possible explanation is lacking from the variation of training dataset, therefore the model learns the relationship between mass and velocity as a commonality of all projectile movements without associating it with projectile characteristics. A possible solution for this is to add more variety to the training dataset, by including different shape of projectiles(*e.g.* a rounded pistol bullet or bullet with different lengths) and more features of the bullet(*e.g.* length and angle of the tail, bearing surface and tip; some bullets are consisted of different materials, and position of centre of mass may vary). To test this

potential enhancement, a more comprehensive data set and a more powerful computer may be necessary; hence, it has been recognized as a limitation of this project and suggested for future research. A hyperparameter tuning process is carried out to select the best combination of parameters for the model:

Population size	RMSE
10	529.243
100	481.195
200	194.918
300	170.979
500	188.324

Table 5.2: Comparison on population size of genetic programming

Generation number	RMSE
10	368.836
50	315.977
100	257.963
200	202.804
300	147.557
500	129.189

Table 5.3: Comparison on generation number of genetic programming

Crossover rate	Mutation rate	RMSE
0.2	0.05	322.184
0.2	0.2	321.959
0.2	0.5	355.103
0.5	0.05	217.924
0.5	0.2	290.742
0.5	0.5	142.738
0.5	0.6	124.062
0.5	0.7	289.887
0.7	0.05	480.868
0.7	0.2	469.208
0.7	0.5	480.868

Table 5.4: Comparison on generation number of genetic programming

Table 5.2 compares the performance of genetic programming model with different size of population while other parameters stay the same(generation=100, crossover rate=0.5,

mutation rate=0.3). The result shows a decline in error as population size increases, which is identical to findings by Gotshall and Rylander[15], which is that there is a higher likelihood that the population will include a chromosome that reflects the optimal solution. An overly large population will not affect the performance, but it could extra time to process the model. Table 5.3 compares genetic programming model with different generation numbers with other parameters fixed. A large generation number means more iterations and the output is closer to the optimal solution after convergence, but it could also have problems. Issues with large generation number is time consumption and overfitting. Since this project is not time-sensitive, a long training time is acceptable. In order to ensure that the estimating function returns a reasonable level of accuracy, I select a large generation number and monitor the RMSE value of model on both training set and validation set. If the model perform way better on training set comparing with validation set, that could mean the model is overfitting. Table 5.4 demonstrates combinations of crossover rates and mutation rates. Hassanat *et al.*[16] suggested a common approach of crossover rate of 0.9 and mutation rate of 0.03, while in this project when both crossover and mutation rate have a value around 0.5 seems to have a relatively good result, which is in contrast to the finding from Hassant *et al.*. By analysing the validation set and selecting the function with the lowest RMSE value, the optimal estimating function is determined. Note that the symbolic regression use `MaxAbsScaler` to scale the variables before training, so each variable needs to be divided by the maximum value of the variable in the training set. The RMSE at each distance when applying the estimating function to the train, validation and test datasets is shown as below:

	RMSE
Training set	103.727
Validation set	93.151
Test set 1	95.692
Test set 2	169.848
Test set 3	326.103

Table 5.5: Symbolic Regression: Overall RMSE on each dataset

Distance from muzzle(Yards)	RMSE
0	0
100	43.514
200	75.961
300	95.140
400	106.151
500	105.587
600	96.448
800	73.048
1000	178.531

Table 5.6: Symbolic Regression: RMSE on the training set at each distance

Distance from muzzle(Yards)	RMSE
0	0
100	195.310
200	70.168
300	27.156
400	59.716
500	73.770
600	70.206
800	34.577
1000	138.403

Table 5.7: Symbolic Regression: RMSE on the validation set at each distance

Distance from muzzle(Yards)	RMSE
0	0
100	169.724
200	37.855
300	48.503
400	95.038
500	110.798
600	109.843
800	64.546
1000	110.815

Table 5.8: Symbolic Regression: RMSE on the test set 1 at each distance

Distance from muzzle(Yards)	RMSE
0	0
100	195.043
200	59.953
300	72.089
400	126.743
500	375.623
600	173.445
800	153.527
1000	44.640

Table 5.9: Symbolic Regression: RMSE on the test set 2

Distance from muzzle(Yards)	RMSE
0	0
100	64.776
200	90.792
300	204.524
400	307.178
500	408.013
600	501.347
800	566.804
1000	263.385

Table 5.10: Symbolic Regression: RMSE on the test set 3

Table 5.5 shows the RMSE value of model on five datasets, including training set from 6.5mm Creedmoor and .260 Remington, validation set from .260 Remington, test set 1 from .260 Remington, test set 2 from .280 Remington and test set 3 from .338 Lapua Magnum. The model has a similar level of accuracy on training set, validation set and test set 1, indicates that the model is not overfitting. The model is less accurate on test set 2 and even worse on test set 3, as the ammunition model in test 2 is more similar than model in test set 3, but still different in size and weight comparing to the training set.
Table 5.6, table 5.7, table 5.8, table 5.9 and table 5.10 shows the performance of symbolic regression model on different data set at different distance. As can be observed, the model performs reasonably well between 200 and 500 yards from the muzzle for all datasets other than test set 3. A less accurate score in test set 3 is understandable given that it contains samples from the .338 Lapua Magnum, which has a more distinct shape compared to the previous sets, which also indicates that this model may not be able to generalised and apply to othe ammunition models.

5.4 Hybrid model for external ballistics

Method of combining equation of motion and symbolic regression model has been discussed in chapter 4. The model is trained bases on the result of session 5.1.1, where the difference of ballistic coefficient estimated velocity and target velocity is considered as the target value. Penalties similar with the symbolic regression model is also applied in this model.

	RMSE
Training set	20.997
Validation set	7.591
Test set 1	11.578
Test set 2	11.813
Test set 3	37.234

Table 5.11: Hybrid model: Overall RMSE on each dataset

Notice that this result is significantly better comparing with the result in table 5.5. The reason this happens is because symbolic regression typically provides high errors as the distance from the muzzle increases, and it might be that the pattern of difference between the real value and the law of motion model is more regular and easier to learn for the symbolic regression than for the velocity value.

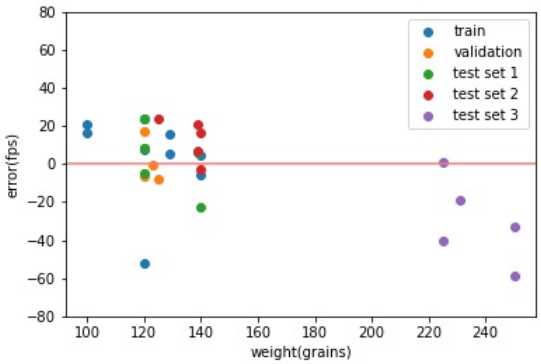


Figure 5.3: Error of velocity prediction of each bullet in different datasets by weight

Table 5.11 presents the RMSE of the model's results after being implemented to five datasets. It is shown that the model generally has a good performance on all datasets in this project. Note that this model performs better on the validation set in comparison to the training set. One possible explanation for this difference in performance is that the training set contains two different types of ammunition, whereas the model may only capture the characteristics of.260 Remington ammunition, which is the type included in both the validation set and test set 1. The prediction result is less accuracy on the test set 3, which

might because of the ammunition model difference, but the error value is still acceptable and could be considered as a usable model. Figure 5.3 is a scatter graph about the performance of the hybrid model on different datasets. The scatters are shown according to the bullet weight and difference between predicted and actual velocity($V_{predict} - V_{real}$). It can be seen that the model generally well predicts all five datasets. The model frequently slightly over-predicts the true value for the train, validation, and first two test sets except for one outlier at weight of 120 grains, whereas for test set 3, the model is generally underestimated and has a greater error value. This result is understandable given that the test set 3 varies more from the training set. Test set 2 also used a different bullet model(.280 Remington) from the training and validating sets, however, the outcome does not seem to have a significant difference.

5.5 Internal ballistics model

Symbolic regression is applied on the simulated data of 6.5mm Creedmoor ammunition from Gordon's Reloading Tool. Models are trained multiple times on different subsets of dataset to test the relation of dataset size and model accuracy. The dataset size selected are 100%, 70%, 40% and 10%. First a normalisation process using `MaxAbsScaler` is applied to the dataset. Any other parameters of symbolic regression such as population size, crossover rate and mutation rate stay the same while training different models.

size of training subset	RMSE
100%	18.657
70%	20.353
40%	56.581
10%	53.858

Table 5.12: RMSE on different size of trainign set

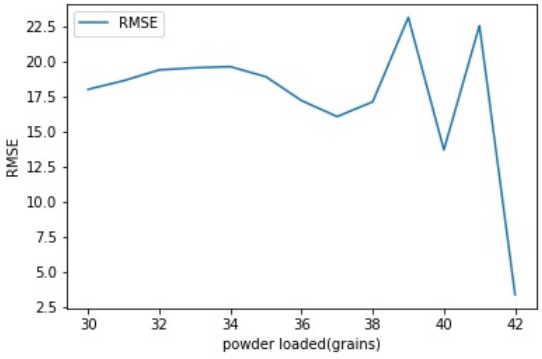


Figure 5.4: RMSE of internal ballistic prediction model on different powder loaded

Table 5.12 shows the RMSE value of applying predicting model trained with different size of training set. It can be seen that in this project, the model using larger dataset has a smaller error. The model using 70% of dataset also generally has a good performance and the RMSE value does not have an significant increase comparing with the full size dataset, so it may worth consider to cut down the size of data by around 30% in training, if the time or equipment is limited. On the other side, it may also indicates that at least 70% of dataset is required to obtain a model with performance comparable to that of the entire dataset. In this project, the model only trained on a small dataset size with around a hundred data, so this conclusion may not able to be generalised. Figure 5.4 shows the RMSE of the model trained using the full size training dataset and applied on the test dataset. The figure demonstrates that the model has a similar level of performance over the range of powder loaded from 30 grains to 41 grains. At the powder weight of 42 grains, I noticed that the RMSE value at 42 grains is considerably smaller than any other weight. By checking the training set and test set, I find out that both training and testing set lack the data of powder loaded weight 42 grains, so the predicting model is not able to learn the behaviour of velocity changing at 42 grains. Moreover, there is only one set of data in the test set that includes the initial velocity at 42 grains, so the small RMSE value at 42 grains cannot represent the performance of the model at this weight.

5.6 Limitation and follow-on works

This project has its limitations:

- **Lack of data:** All the datasets for external ballistics are obtained from the book Ammo & Ballistics 6[14], and the data for variables "boattail", "roundtip", "cannelure" are from the relavent bullet product pages, dataset for internal ballistics are generated using Gordons Reloading Tool[2]. Since each piece of data is manually entered, collecting a sizable dataset is difficult for me.
  Moreover, for the external ballistics predicting model, the bullet specifications are not sufficiently precise. A bullet cannot be adequately described by its weight, ballistic coefficient, and a few additional features like tail and tip shapes. Typically, a sufficient dataset and enough features are needed in order to build an accurate machine learning model.

- **Limitation of genetic programming:** Genetic programming has its drawbacks. As the evolution process is completely random, genetic programming cannot guarantee to return the optimal performance model. This suggests that there may be a solution superior to the prediction functions claimed in this report. Due to this randomness, it is also difficult to ensure that the algorithm would provide relatively decent results within certain generations, hence the training duration cannot be specified. It may be possible to improve the performance of genetic programming models by combining it with other algorithms.

- **Other target variables that have not been considered:** Other than velocity, factors like the bullet decline route and wind drift are also taken into consideration during the phase of project planning. These variables are equally, if not more important, than the velocity when describing the movement of projectiles. However, it is difficult for symbolic regression using DEAP to compute numerous outputs simultaneously, and it is also hard for my personal computer to perform the computations. These variables are not considered anymore in the experiment stage.

The following is a list of the identified further work:

- Use a larger, more comprehensive dataset for training. As stated in Chapter 1, one of the advantages of symbolic regression comparing with other trajectory prediction methods is the ability to examine the impact of different bullet attributes on bullet trajectory. For instance, if the parameter preceding the cannelure variable is minor, we may conclude that the cannelure design has little effect on bullet velocity.

- Use a more powerful computer to train the algorithm, so that the model can be trained on a larger and more complete data set and achieve greater accuracy. The theoretical method to velocity could also be estimated with a more comprehensive equation of motion and a narrower gap for a more precise answer.

- Improvement on genetic programming algorithm. Given that a genetic algorithm may not always produce the optimal outcome, it may be possible to combine genetic programming with other techniques to improve the outcome, for example Mundhenk *et al.*[32], which is an RNN approach of symbolic regression in which genetic programming is used between each layer, evaluating output samples and running a number of genetic programming generations. The samples generated by genetic programming and the original samples from RNN are then mixed and utilised to train RNN.

Chapter 6

Conclusions

This project aimed to find a good prediction model for ballistics. The methods chosen for predicting the velocity of bullets at different distances from the muzzle are a theoretical approach by the equation of motion, symbolic regression through genetic programming with DEAP framework, and a hybrid model that uses a combination of the above two. The models are trained on and aimed to fit datasets from book Ammo & Ballistics 6[14]. In order to achieve the requirement of predicting a trajectory, penalty is added for different situations. The requirement of this project is able to be achieved, and the performance of models are evaluated using a standardised measurement(RMSE). The evaluation and discussion about the results are included in chapter 5. Each method in chapter 5 is trained multiple times and the best performance model is selected. When the distance from the muzzle is between 200 and 600 yards, a model based simply on symbolic regression has a good accuracy. However, it usually performs poorly when the distance is either short (100 yards) or long (1,000 yards). The explanation may be that it is difficult to comprehend the law of motion and the pattern of velocity using solely symbolic regression, and the periodic operators(*e.g.* sin, cos) add some periodicity to the expression, so the error does not vary consistently with distance. The hybrid model has the best outcome among three models. It is possible that the pattern of difference between the real value and the model of the law of motion is more regular and easier to learn for the symbolic regression than for the velocity value.

This project also trains a symbolic regression model on internal ballistics. There is few innovation involved, and the procedure and outcomes are straightforward. A model is trained using various training set sizes in an experiment, and the results show that the training set size may be reduced without adversely impacting the training outcomes in terms of time or hardware requirements. It may also indicate that for a relatively good performance model to be obtained, at least 70% of data from training set need to be kept according to table 5.12.

It is suggested that future research on this topic utilise a higher-performing training machine and a larger, more comprehensive training dataset, and that additional target variables, such as path and wind drift, be considered. This research uses genetic programming for symbolic regression as it is the most common approach to symbolic regression; yet, genetic programming could possibly be improved by integrating it with different algorithms to build

an optimal model for symbolic regression.

Bibliography

[1] Deap documentation. https://deap.readthedocs.io/en/master/. Accessed: 11-08-2022.

[2] Gordons Reloading Tool Community: User manual Handbook. https://www.grtools.de/doku.php?id=en:doku:start. Accessed: 30-08-2022.

[3] Symbolic regression problem: Introduction to gp. https://deap.readthedocs.io/en/master/examples/gp_symbreg.html. Accessed: 22-08-2022.

[4] Summary of federal firearms laws. OFFICE OF THE UNITED STATES ATTORNEY DISTRICT OF MAINE, Sep 2010.

[5] Ahsan, M. M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., and Siddique, Z. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies 9*, 3 (Sep 2021).

[6] Alexander, J. H. *Universal dictionary of weights and measures, ancient and modern, reduced to the standards of the United States of America.* Baltimore, W. Minifie and Co., 1850.

[7] Braun, W. F. Aerodynamic data for small arms projectiles, 1973.

[8] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122.

[9] Bullets, S. How the ballistic coefficient is measured by firing tests. https://www.sierrabullets.com/exterior-ballistics/4-3-how-the-ballistic-coefficient-is-measured-by-firing-tests/. Accessed: 18-08-2022.

[10] Charters, A. C., and Thomas, R. N. The aerodynamic performance of small spheres from subsonic to high supersonic velocities. *Journal of the Aeronautical Sciences 12*, 4 (Oct 1945), 468–476.

[11] Cobb, H. G. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments.

[12] Courtney, M., and Courtney, A. The truth about ballistic coefficients.

[13] Dekking, M. *A modern introduction to probability and statistics*. London : Springer, 2005.

[14] Forker, B. *Ammo Ballistics 6*. Safari Press, 2017.

[15] Gotshall, S., and Rylander, B. Optimal population size and the genetic algorithm.

[16] Hassanat, A., Almohammadi, K., Alkafaween, E., Abunawas, E., Hammouri, A., and Prasath, V. B. S. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information 10*, 12 (2019).

[17] Hitchcock, H. P. Aerodynamic Data for Spinning Projectiles, 1947.

[18] Hornady. Hornady Ballistic Calculators. https://www.hornady.com/team-hornady/ballistic-calculators/#!/. Accessed: 20-08-2022.

[19] Hyndman, R. J., and Koehler, A. B. Another look at measures of forecast accuracy. *International Journal of Forecasting 22*, 4 (2006), 679–688.

[20] Jin, Y., Fu, W., Kang, J., Guo, J., and Guo, J. Bayesian symbolic regression, 2019.

[21] Joyner, D., Čertík, O., Meurer, A., and Granger, B. E. Open source computer algebra systems: Sympy. *ACM Commun. Comput. Algebra 45*, 3/4 (jan 2012), 225–234.

[22] Kim, J. T., Kim, S., and Petersen, B. K. An Interactive Visualization Platform for Deep Symbolic Regression. *Conference on Artificial Intelligence* (Jul 2020), 5261–5263.

[23] Koza, J. R. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing 4*, 2 (Jun 1994), 87–112.

[24] La Cava, W., Orzechowski, P., Burlacu, B., de França, F. O., Virgolin, M., Jin, Y., Kommenda, M., and Moore, J. H. Contemporary symbolic regression methods and their relative performance, 2021.

[25] Ladommatos, N. Drag coefficients of air rifle pellets with wide range of geometries. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 235*, 21 (Nov 2021), 5365–5384.

[26] Ladommatos, N. Influence of air rifle pellet geometry on aerodynamic drag. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology 235*, 4 (Dec 2021), 257–276.

[27] Lahti, J., Sailaranta, T., Harju, M., and Virtanen, K. Control of exterior ballistic properties of spin-stabilized bullet by optimizing internal mass distribution. *Defence Technology 15*, 1 (2019), 38–50.

[28] Lobo, F. G., Lima, C. F., and Michalewicz, Z. *Parameter Setting in Evolutionary Algorithms*. Heidelberg: Springer, 2007.

[29] LU, Z., and HU, W. Estimation of ballistic coefficients of space debris using the ratios between different objects. *Chinese Journal of Aeronautics 30*, 3 (2017), 1204–1216.

[30] McCoy, R. L. The Aerodynamic Characteristics of .50 Ball, M33, API, M8, and APIT, M20 Ammunition, 1990.

[31] Mishchenko, E. G. What causes bullet's wind drift and how significant is it in pistol shooting?

[32] Mundhenk, T. N., Landajuela, M., Glatt, R., Santiago, C. P., Faissol, D. M., and Petersen, B. K. Symbolic Regression via Neural-Guided Genetic Programming Population Seeding, 2021.

[33] MyCoy, R. L. *Modern Exterior Ballistics*. Schiffer Publishing Ltd, 2012.

[34] National Oceanic and Atmospheric Administration. Speed of sound calculator. https://www.weather.gov/epz/wxcalc_speedofsound. Accessed: 12-08-2022.

[35] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[36] Petch, J., Di, S., and Nelson, W. Opening the black box: The promise and limitations of explainable machine learning in cardiology. *Canadian Journal of Cardiology 38*, 2 (Feb 2022), 204–213.

[37] Petersen, B. K., Larma, M. L., Mundhenk, T. N., Santiago, C. P., Kim, S. K., and Kim, J. T. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations* (2021).

[38] Rafeie, M., and Teymourtash, A. Aerodynamic and dynamic analyses of three common 4.5 mm-caliber pellets in a transonic flow. *Scientia Iranica 23* (Oct 2016), 1767–1776.

[39] Rainville, F.-M. D., Fortin, F.-A., Gardner, M.-A., Parizeau, M., and Christian Gagne, booktitle=GECCO '12, y. Deap: a python framework for evolutionary algorithms.

[40] Salimipour, S. E., Teymourtash, A. R., and Mamourian, M. Investigation and comparison of performance of some air gun projectiles with nose shape modifications. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology 233*, 1 (Mar 2019), 3–15.

[41] Selin, H. *Encyclopaedia of the History of Science, Technology, and Medicine in Non-Westen Cultures*. Springer Netherlands, 1997.

[42] sellier bellot. Ballistic coefficient calculation. https://www.sellier-bellot.cz/en/products/ballistic-coefficient-calculation/. Accessed: 18-08-2022.

[43] Team, S. D. SymPy 1.10.1 documentation. https://docs.sympy.org/latest/index.html#. Accessed: 23-08-2022.

[44] U.S. Marine Corps. Tactics, techniques, and procedures for the field artillery manual cannon gunnery, 1996.

[45] Weinacht, P., Newill, J., and Conroy, P. Conceptual design approach for small-caliber aeroballistics with application to 5.56-mm ammunition. 74.

[46] Wong, G. S. K., and Embleton, T. F. W. Variation of the speed of sound in air with humidity and temperature. *The Journal of the Acoustical Society of America 77*, 5 (May 1985), 1710–1712.

Appendices

Appendix A

Display of result from symbolic regression model

Symbolic regression model trained from the training set normalised with `MaxAbsScaler` in session 4.3 is demostrated as below:

$$D = \text{distance from muzzle(yards)}/1000$$
$$BC = \text{ballistic coefficient}/0.585$$
$$Weight = \text{weight(grains)}/140$$
$$IV = \text{initial velocity at muzzle(feet per second)}/3200$$
$$V = \text{velocity(feet per second)}/3200$$

And function

```
V=add(mul(D, add(add(neg(mul(protectedDiv(IV, add2(add(sin(div2(BC)), add(add(div2(BC),
protectedDiv(D, protectedDiv(protectedDiv(add(BC, cos(BC)), BC), mul(mul2(div2(add(IV,
BC)), D), protectedDiv(D, sin(protectedDiv(roundtip, roundtip)))))))), Weight)),
Weight)), add2(div2(add2(mul(IV, protectedDiv(cos(mul(D, add(BC, D))), D)),
mul(protectedDiv(mul(mul(roundtip, add2(sub(neg(IV), protectedDiv(neg(roundtip),
IV)), IV)), cos(Weight)), add2(add2(mul2(mul(Weight, div2(neg(Weight))),
cos(scala2(boattail))), div2(D)), add(add2(BC, add2(add2(IV, Weight), div2(BC))),
D))), IV))), Weight))), neg(D)), cos(cos(mul(Weight, D)))))), IV)
```

This function is then converted to human-readable form using `SymPy`.

$$V = -D\left(D + \frac{IV\left(\frac{IV^2 roundtip\left(IV + \frac{roundtip}{IV}\right)}{6BC+3D+4IV-Weight^2\cos(2boattail)+8Weight} + 2Weight + \frac{IV\cos(D(BC+D))}{2D}\right)}{\frac{2BCD^3\left(\frac{BC}{2}+\frac{IV}{2}\right)}{(BC+\cos(BC))\sin(1)} + \frac{BC}{2} + 3Weight + \sin\left(\frac{BC}{2}\right)} - \cos(\cos(\cos(DWeight)))\right) + IV \tag{A.1}$$

For each fraction in the equation, if the denominator equals to zero, the whole fraction should equal to one. This is set in the function `protectDiv( )`, but it is not presented in the latex form maths expression.

Appendix B

Display of result from hybrid model

This appendix shows the result of the hybrid model described in session 4.4. The training set is normalised with `MaxAbsScaler`.

D = distance from muzzle(yards)/1000
BC = ballistic coefficient/0.585
$Weight$ = weight(grains)/140
IV = initial velocity at muzzle(feet per second)/3200
$Veom$ = velocity calculated from equation of motion(feet per second)/3200
V_{gap} = velocity difference of equation of motion and target(feet per second)/3200

And function:

```
V_gap=mul(mul(IV, Veom), mul(sub(sub(protectedDiv(mul(-1, protectedDiv(mul(BC,
add2(add(mul(D, sub(sub(Weight, cannelure), mul(Veom, mul2(Veom, mul2(sub(IV,
Veom), boattail))))), Veom), mul(D, protectedDiv(sub(scala2(Weight), sin(mul2(Veom,
Veom))), mul(scala2(mul2(protectedDiv(mul2(IV, 0), cos(-1)), roundtip)),
mul2(scala2(protectedDiv(protectedDiv(BC, boattail), boattail)), mul(mul2(BC,
Weight), mul2(IV, boattail)))))))), mul(scala2(add2(sub(Weight, sub(D, cannelure)),
mul(D, protectedDiv(sub(scala2(mul2(roundtip, D)), Weight), mul(scala2(mul2(cos(roundtip),
mul2(Veom, boattail))), mul2(scala2(div2(boattail)), mul2(add(Veom, cos(boattail)),
add(sub(D, D), Weight)))))))), mul2(scala2(boattail), roundtip)))), mul(scala2(Weight),
cos(sin(mul2(Veom, mul2(sub(IV, Veom), mul2(BC, IV)))))), cos(sub(Weight, sub(D,
cannelure)))), boattail), sub(IV, Veom)))
```

This function is then converted to human-readable form using `SymPy`.

(B.1)

$$V = IVVeom(IV - Veom)*$$

$$\left(-\frac{2BCD(Veom+D(Weight-cannelure-4boattailVeom^2(IV-Veom)))}{8boattail*roundtips(Weight-D_cannelure+D\frac{4roundtipD-Weight}{32cos(roundtip)Veomboattail^2(Veom+cos(boattail))Weight}}\right) - cos(Weight - D + cannelure) - boattail))$$

$$2Weightcos(sin(2Veom(4BCIV(IV - Veom))$$

For each fraction in the equation, if the denominator equals to zero, the whole fraction should equal to one. This is set in the function `protectDiv( )`, but it is not presented in the latex form maths expression. The estimation result of the hybrid model should be:

$$V = Veom + V_{gap} * 144.058 \tag{B.2}$$

where 144.058 is the maximum value of velocity gap in the training set.